

# Optimizing Budget Allocation in Graphs

Boaz Ben-Moshe\*

*Department of Computer Science and Mathematics, Ariel University, Ariel, Israel.*

Michael Elkin<sup>1</sup>

*Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel.*

Lee-Ad Gottlieb

*Department of Computer Science and Mathematics, Ariel University, Ariel, Israel.*

Eran Omri

*Department of Computer Science and Mathematics, Ariel University, Ariel, Israel.*

---

## Abstract

In a classical facility location problem we consider a graph  $G$  with fixed weights on the edges of  $G$ . The goal is then to find an optimal positioning for a set of facilities on the graph with respect to some objective function. We introduce a new framework for facility location problems, where the weights on the graph edges are not fixed, but rather should be assigned. The goal is to find the valid assignment for which the resulting weighted graph optimizes the facility location objective function. We present algorithms for finding the optimal *budget allocation* for the center point problem and for the median point problem on trees. Our algorithms work in linear time, both for the case that a candidate vertex is given as part of the input, and for the case where finding a vertex that optimizes the solution is part of the problem. We also present a hardness result on the general case of the center point problem, followed by an  $O(\log^2(n))$  approximation algorithm for the general case - on general metric spaces.

*Keywords:* Facility Location, Graph Radius, Budget Graphs

---

## 1. Introduction

A typical facility location problem has the following structure: the input includes a weighted set  $D$  of demand locations, a set  $F$  of feasible facility locations, and a distance function  $d$  that measures the cost of travel between a pair of locations. For each  $F' \subseteq F$ , the quality of  $F'$  is determined by some underlying objective function ( $obj$ ). The goal is to find a subset of facilities  $F' \subseteq F$ , such that  $obj(F')$  is optimized (maximized or minimized). One important class of facility location problems is the *center point*, in which the goal is to find one facility in  $F$  that minimizes the maximum distance between

---

\*Corresponding author

*Email addresses:* benmo@g.ariel.ac.il (Boaz Ben-Moshe), elkinm@cs.bgu.ac.il (Michael Elkin), leead@ariel.ac.il (Lee-Ad Gottlieb), omrier@ariel.ac.il (Eran Omri)

<sup>1</sup>This author's research has been supported by the Binational Science Foundation, grant No. 2008390.  
*Preprint submitted to Theoretical Computer Science June 10, 2014*

a demand point and the facility. Henceforth, we refer to this distance as *graph radius*. In another important class of problems, *graph median*, the goal is to find the facility in  $F$  that minimizes the average distance (i.e., the sum of the distances) between a demand point and the facility. In this paper we consider a new model for facility location on graphs, for which both problems are addressed.

### 1.1. The New Model

This paper<sup>2</sup> suggests a new model for budget allocation problems on weighted graphs. The new model addresses optimization problems of allocating a fixed budget onto the graph edges where the goal is to find a subgraph that optimizes some objective function (e.g., minimizing the graph radius). Problems such as center point and median point on trees and graphs have been studied extensively [4, 6, 8, 9]. Yet, in most cases the input for such problems consists of a given (fixed) graph. Motivated by well-known budget optimization problems [1, 3, 5, 10] raised in the context of communication networks, we consider the graph to be a communication graph, where the weight of each edge (link) corresponds to the delay time of transferring a (fixed length) message over the link. We suggest a *Quality of Service* model for which the weight of each edge in the graph depends on the budget assigned to it. In other words, paying more for a communication link decreases its delay time.

More formally, we consider an undirected graph  $G = \langle V, E \rangle$  equipped with a length function  $\ell(\cdot)$  on the edges. The weighted graph  $(G, \ell)$  induces a metric space. i.e., the length function  $\ell(\cdot)$  satisfies the triangle inequality. Let  $B$  be a positive value, which we call a *budget parameter*. In our model, allocating a budget  $\mathcal{B}(e) \geq 0$  to an edge  $e \in E$  of length  $\ell(e)$  means that the *weight* of  $e$  will be  $w(e) = w_{\mathcal{B}}(e) = \frac{\ell(e)}{\mathcal{B}(e)}$  (in case  $\mathcal{B}(e) = 0$ , we let  $w(e) = +\infty$ ).

The distance between two vertices  $v$  and  $u$  in  $(G, \mathcal{B})$  is the sum of weights on the edges of a minimal weighted path between them, denoted as  $\delta_{\mathcal{B}}(v, u)$ . The *radius* of  $(G, w)$  with respect to a designated vertex  $r$   $\text{Rad}_r(G, w)$  is the maximum distance (with respect to the underlying allocation  $\mathcal{B}(\cdot)$ ) between  $r$  and any other vertex  $v$  in the graph. In the *rooted budget radius* problem, we are given a graph  $G = \langle V, E \rangle$ , equipped with a length function  $\ell(\cdot)$ , a budget parameter  $B$  and a designated vertex  $r$ . The goal is to assigned none-negative values  $\mathcal{B}(e)$  to the edges of  $G$  that sum up to  $B$  (i.e.,  $\sum_{e \in E} (\mathcal{B}(e)) = B$ ), and so that the rooted radius of the graph  $(G, w)$  with  $w = \frac{\ell}{\mathcal{B}}$  (for some positive constant  $c$ ; in this paper we mainly consider  $c$  to be 1), with respect to the vertex  $r$  is minimized.

In the *unrooted budget radius* problem, we are given a graph  $G = \langle V, E \rangle$ , equipped with a length function  $\ell(\cdot)$ , and a budget parameter  $B$ . The goal is to minimize the rooted budget radius problem over all choices of  $r \in V$ . That is, our goal is to find a vertex  $r$  and a corresponding assignment  $\mathcal{B}(e)$  that minimizes the budget radius problem with respect to  $G$ ,  $B$ , and  $r$ . The *budget diameter* problem is defined analogously, where the goal is to minimize the maximum distance between any pair of vertices in  $(G, w)$ .

In the *budget median* problem, we are given a graph  $G = \langle V, E \rangle$ , equipped with a length function  $\ell(\cdot)$ , and a budget parameter  $B$ . The goal is to find a vertex  $r$  and a corresponding assignment  $\mathcal{B}(e)$  that minimize the *average distance* between  $r$  and other vertices of the graph, defined as  $\frac{1}{n} \cdot \sum_{v \in V} \delta_{\mathcal{B}}(r, v)$ . The vertex  $r$  that minimizes the budget median problem is called the *budget median*.

### 1.2. Motivation

We were motivated by communication optimization problems in which for a fixed 'budget' one needs to design the 'best' network layout. The quality of service (*QoS*) of a link between two nodes depends on two main factors: i) The distance between the nodes. ii) The infra-structure of the link (between the two nodes). While the location of the nodes is often fixed and cannot be changed, the infra-structure type and service can be upgraded - it is a price-dependent service.

Quality of service is related to different parameters like, bandwidth, delay time, jitter, packet error rate and many others. Given a network graph, the desired objective is to have the best *QoS* for a given (fixed) budget. In this paper we focus on minimizing the maximum and the average delay time using a fixed budget.

### 1.3. Related Work

The problems of Center Point, Median Point on graphs (networks) have been studied extensively, see [4, 8] for a detailed surveys on *facility location*. There are various optimization problems dealing with finding the best graph; A typical *graph or network improvement* problem considers a graph which needs to be improved by adding the smallest number of edges in order to satisfy some constraint (e.g., maximal radius), see [1, 2, 5, 10]. Spanner graph problems [7] consider what can be seen as the *inverse* case of *network improvement problems*. In a typical spanner problem we would like to keep the smallest subset of edges from the original graph while maintaining some constraint. See [7] for a detailed survey on spanners. Observe that both *network improvement* and *spanner graph* problems can be modeled as a discrete version of our suggested new model.

### 1.4. Our Contribution

In this paper we present linear time algorithms for rooted and unrooted budget radius and budget median problems on trees. We also prove that the general version of the problem is N.P.-hard and devise an  $O(\log^2(n))$  approximation algorithm for the budget radius problem on general metric spaces.

## 2. Preliminaries

In this section we introduce basic notations and definitions that are used for describing the suggested budget graph framework. Let  $G = \langle V, E \rangle$  be a graph with some length function  $\ell : E \mapsto \mathbb{R}^+$ . A *valid budget allocation*  $\mathcal{B}(\cdot)$  to  $E$  is a non-negative real function, such that  $\sum_{e \in E} \mathcal{B}(e) = 1$  (here and in the rest of the paper we assume that the total budget  $B$  equals 1; this is without loss of generality since an optimal solution with budget of 1 is easily scaled to any budget  $B$ ). Let  $E = \{e_1, \dots, e_{|E|}\}$ . We denote  $b_i \stackrel{\text{def}}{=} \mathcal{B}(e_i)$ , and for every  $E' \subseteq E$  we denote  $\mathcal{B}(E') = \sum_{e_i \in E'} b_i$ . Given a valid budget allocation  $\mathcal{B}$  to  $E$ , the *weight* of an edge  $e \in E$ , denoted  $\omega_{\mathcal{B}}(e)$ , is a function of  $\ell(e)$  and  $\mathcal{B}(e)$ . Throughout this paper we consider the case where  $\omega_{\mathcal{B}}(e) \stackrel{\text{def}}{=} \frac{\ell(e)}{\mathcal{B}(e)}$ .

**Definition 2.1** (weighted distance). *Let  $u, v \in V$  be two vertices. The weighted distance between  $v$  and  $u$ , denoted  $\delta_{\mathcal{B}}(v, u)$ , is the minimum weight over all simple paths between  $u$  and  $v$ . Namely,  $\delta_{\mathcal{B}}(v, u) \stackrel{\text{def}}{=} \min(\{\sum_{e \in P} \omega_{\mathcal{B}}(e) : P \text{ is a simple path from } v \text{ to } u\})$ .*

Table 1 summarizes the above notations. In the rest of this section we present two specific facility location problems in the budget graph framework. In Section 2.1, we describe the budget radius problem and in Section 2.2, we describe the budget median problem.

Notation	Explanation
$G = \langle V, E \rangle$	a general undirected graph (induced by some metric space)
$\ell(e)$	the (a priori) length of an edge $e \in E$ .
$\mathcal{B}(e)$	the budget fraction allocated to $e \in E$ .
$\mathcal{B} = \{b_1, \dots, b_{ E }\}$	an alternative notation for the function $\mathcal{B}$ .
$\omega(e) = \frac{\ell(e)}{\mathcal{B}(e)}$	the (budget implied) weight of $e \in E$ .
$(G, w) = G(\mathcal{B})$	the <i>budget-graph</i> implied by an allocation $\mathcal{B}$
$\delta_{\mathcal{B}}(v, u)$	the distance between two vertices in $G(\mathcal{B})$ .

**Table 1:** Notations that are used throughout the paper to present the new budget graph model.

### 2.1. The Budget Radius Problem

We next introduce some definitions and notations to define the setting of the *budget radius* problem on graphs. In the first setting we consider, a candidate center node to the graph is given and the goal is to find an optimal budget allocation that minimizes the radius of graph with respect to the resulting distances between nodes in the graph (the weighted radius). In a second setting, a candidate center node is not given and the goal is to find both a center node and a budget allocation that are together optimal with respect to the weighted radius. The formal definitions follow.

**Definition 2.2** (rooted weighted radius). *Given a valid budget allocation  $\mathcal{B}$  to  $E$  and a vertex  $r \in V$ , the weighted radius of  $G$  with respect to  $r$  is defined as  $\text{wr}_{\mathcal{B}}(r) = \text{wr}_{\mathcal{B}}(G, r) \stackrel{\text{def}}{=} \max_{v \in V}(\delta_{\mathcal{B}}(r, v))$ .*

Given a graph  $G = \langle V, E \rangle$  (induced by some metric) and a node  $r \in V$ , we next define what it means for an allocation to be optimal with respect to the budget radius problem. We usually omit  $G$  from the notation whenever it can be derived from the context.

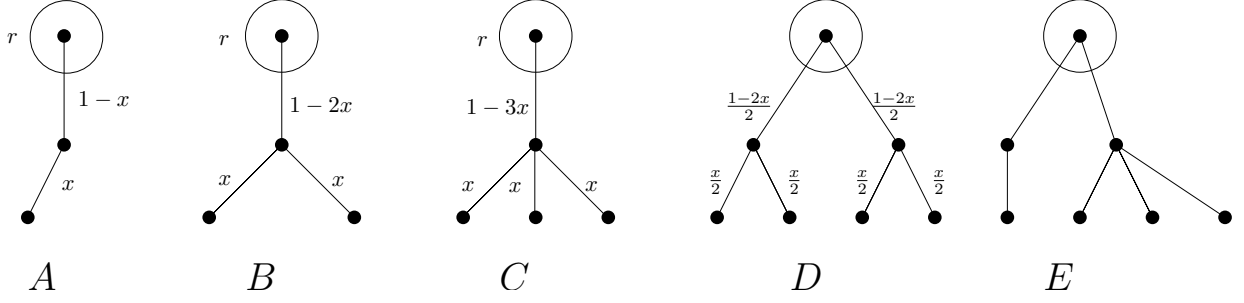
**Definition 2.3** (rooted budget radius). *An optimal allocation for  $(G, r)$  with respect to the budget radius problem is a valid allocation for which the weighted radius with respect to  $r$  is minimized. There may be several optimal allocations. We take an arbitrary optimal allocation and denote it by  $\mathcal{B}_r^* = \mathcal{B}_r^*(G)$ . We further refer to this allocation as the optimal allocation for  $(G, r)$ .*

*The budget radius of  $G$  with root  $r$ , denoted  $\text{BR}(r) = \text{BR}(G, r)$ , is the weighted radius of  $G$  with respect to  $r$  and  $\mathcal{B}_r^*$ , i.e.,  $\text{BR}(r) = \text{wr}_{\mathcal{B}_r^*}(r)$ .*

We next give the appropriate definitions for the setting where the root is not given as part of the input to the problem. We call this problem the *unrooted* budget radius problem. The general setting, where the goal is to find a node in the graph that minimizes the radius is sometimes referred to as the center point problem.

**Definition 2.4** (unrooted budget radius). *The budget radius of a graph  $G$  is the value  $\text{BR} = \text{BR}(G) \stackrel{\text{def}}{=} \min_{v \in V} \text{BR}(G, v)$ . We refer to a pair  $(\mathcal{B}^*, r^*)$  as the optimal allocation for  $G$  (in the unrooted setting) if  $\mathcal{B}^*$  is a valid allocation to  $E$  and  $r^* \in V$  is the vertex with the smallest corresponding radius, i.e.,  $\text{wr}_{\mathcal{B}^*}(r^*) = \text{BR}$ .*

We demonstrate the above definitions using the few examples presented in Figure 1.



**Figure 1:** Few simple examples of a budget graph radius problem on rooted trees (with a given center  $r$ ). Assume that each edge  $e$  has length  $\ell(e) = 1$ . The total budget is 1 and the optimal budget allocation of each edge is shown as a function of  $x$ . Consider case  $B$ : the optimal solution minimizes the following function:  $f(x) = \frac{1}{1-2x} + \frac{1}{x}$ . Note that in cases where  $x$  equals  $\frac{1}{3}$  or  $\frac{1}{4}$  the radius is 6, while the optimal allocation of  $x$  is approximately 0.293, and the radius is approximately 5.828. In case  $C$  the function is:  $f(x) = \frac{1}{1-3x} + \frac{1}{x}$ . For which  $x$  is approximately 0.211, and the radius is approximately 7.464. Case  $D$  is composed from two cases of  $B$  therefore the radius is twice the budget radius of  $B = 11.656$ . Case  $E$  is composed from cases  $A, C$  therefore the radius is the sum of the two radii = 11.464.

## 2.2. The Budget Median Problem

Given a graph, the general median problem is defined as follows: find a node ( $m^*$ ) in graph from which the sum of all weighted distances to all other nodes in the graph is minimized. On *budget graphs*, the *budget median* problem is to find a median node and a corresponding optimal allocation which minimizes the sum of distances between the median node and all the other nodes in the graph (with respect to the budget allocation).

**Definition 2.5** (rooted weighted budget median). *Given a valid budget allocation  $\mathcal{B}$  to  $E$  and a vertex  $m \in V$ , the weighted median of  $G$  with respect to  $m$  is defined as  $\text{wm}_{\mathcal{B}}(m) = \text{wm}_{\mathcal{B}}(G, m) \stackrel{\text{def}}{=} \sum_{v \in V} (\delta_{\mathcal{B}}(m, v))$ .*

Given a graph  $G = \langle V, E \rangle$  (induced by some metric) and a node  $m \in V$ , we next define what it means for an allocation to be optimal with respect to the budget median problem.

**Definition 2.6** (rooted budget median). *An optimal allocation for  $(G, m)$  with respect to the budget median problem is a valid allocation for which the sum of all weighted distances to  $m$  is minimized. There may be several optimal allocations. We take an arbitrary optimal allocation and denote it by  $\mathcal{B}_m^* = \mathcal{B}_m^*(G)$ . We further refer to this allocation as the optimal allocation for  $(G, m)$ .*

*The budget median of  $G$  with root  $m$ , denoted  $\text{BM}(m) = \text{BM}(G, m)$ , is the weighted sum of distances from  $m$  to all other nodes of  $G$  with respect to  $\mathcal{B}_m^*$ , i.e.,  $\text{BM}(m) = \text{wm}_{\mathcal{B}_m^*}(m)$ .*

We next give the appropriate definitions for the setting where the root is not given as part of the input to the problem. We call this problem the *unrooted* budget median problem.

**Definition 2.7** (unrooted budget median). *The budget median of a graph  $G$  is the value  $\text{BM} = \text{BM}(G) \stackrel{\text{def}}{=} \min_{v \in V} \text{BM}(G, v)$ . We refer to a pair  $(\mathcal{B}^*, m^*)$  as the optimal allocation for  $G$  (in the unrooted setting) if  $\mathcal{B}^*$  is a valid allocation to  $E$  and  $m^* \in V$  is the vertex with the smallest corresponding sum of distances to all other nodes in the graph, i.e.,  $\text{wm}_{\mathcal{B}^*}(m^*) = \text{BM}$ .*

### 3. The Budget Radius Problem for Trees

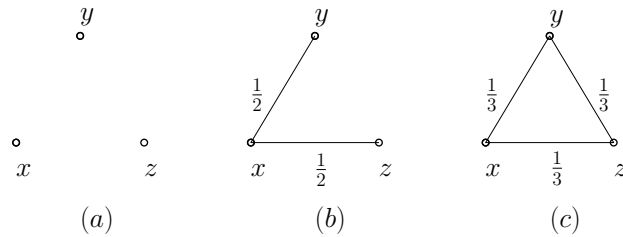
In this section, we solve the budget radius problem for trees. We start by showing that the solution to the budget radius problem for a general graph  $G$  is always in the form of a tree spanning  $G$ . In Section 3.1, we solve the rooted budget radius problem for trees. In Section 3.2, we turn to the unrooted variant of the problem.

**Lemma 3.1.** *Let  $G = \langle V, E \rangle$  be a connected graph and let  $\ell$  be a length function on  $E$ . An optimal budget allocation (with respect to the budget radius problem)  $(\mathcal{B}^*, r^*)$  for  $G$ , has the property that  $H = \langle V, E_{\mathcal{B}^*} \rangle$ , where  $E_{\mathcal{B}^*} = \{e_i \in E : b_i^* > 0\}$  is a tree spanning  $G$ .*

*Proof.* First, assume towards a contradiction that  $H$  is not connected. Then, for any center vertex  $r^*$ , there is a vertex  $v \in V$  such that there is no path in  $H$  connecting  $r^*$  to  $v$ . In this case, however, any path between  $r^*$  and  $v$  in  $(G, w) = G(\mathcal{B}^*)$  contains an edge of infinite weight. Hence,  $\delta_{\mathcal{B}^*}(r^*, v) = \infty$ , and so the radius implied by the budget allocation  $\mathcal{B}^*$  is  $\infty$ . This means that  $\mathcal{B}^*$  is not optimal, since there are allocations yielding a finite radius, e.g., the uniform allocation, assigning an equal portion of the budget to each edge (i.e.,  $\mathcal{B}(e) = \frac{1}{|E|}$  for all  $e \in E$ ).

Second, assume towards a contradiction that  $H$  contains a cycle. Let  $H'$  be the shortest path tree rooted at  $r^*$ . Hence, there exists at least one edge  $e \in E_H \setminus E_{H'}$ . By definition,  $\mathcal{B}^*(e) > 0$ . We define a new budget allocation  $\mathcal{B}'$  distributing the budget portion allocated to  $e$  evenly among all edges in  $H'$ . It is easy to verify that the budget radius of  $\mathcal{B}'$  is strictly smaller than that of  $\mathcal{B}^*$ . This is a contradiction to the optimality of  $\mathcal{B}^*$ . Hence,  $H$  is a spanning tree.  $\square$

An analogous argument applies to the budget median problem. We note, however, that the above is not true for the budget diameter problem (see figure 2).



**Figure 2:** (a) Three points in the plane:  $x, y, z$  are the nodes of a unite equilateral triangle. (b) Tree: optimal radius (2), non optimal diameter (4). (c) Cycle graph: non-optimal radius (3), optimal diameter (3).

The above lemma suggests that it is interesting to consider the Budget Radius problem for the subclass of trees. In the sequel, we present an algorithm solving this problem. We first consider the case where a designated center node  $r$  is given as a part of the input, and an optimal budget allocation  $\mathcal{B}^*$  is sought. We use the standard terminology and

refer to  $r$  as the root of the tree (rather than, the center). Thereafter, we consider the general case in trees, where the problem is to find a pair  $(\mathcal{B}^*, r^*)$  minimizing the budget radius of the tree.

### 3.1. The Budget Radius for Rooted Trees

We next consider two sub-families of rooted trees that will later be the basis for our recursive construction of an optimal valid budget allocation to the edges of general trees. First, we consider a tree in which the root has just one child.

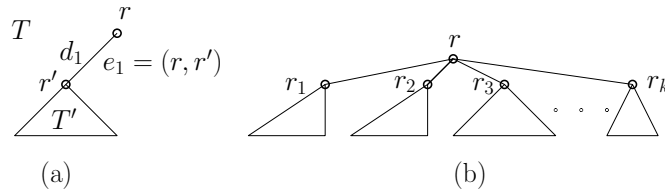
**Lemma 3.2.** *Let  $T$  be a tree rooted at  $r$ , with some length function  $\ell$  on the edges of  $T$ . Assume  $r$  has a single child  $r'$  (the root of the subtree  $T'$ ), and let  $R' = \text{BR}(T', r')$  and  $q = \ell(r, r')$ . Then, an optimal budget allocation  $\mathcal{B}^*$  assigns to the edge  $e = (r, r')$  a fraction  $\beta = \frac{\sqrt{q}}{\sqrt{R'} + \sqrt{q}}$ . It follows that  $\text{BR}(T, r) = \frac{q}{\beta} + \frac{R'}{1-\beta}$ .*

*Proof.* Let  $E$  be the set of edges in  $T$  and  $E' = E \setminus \{e\}$  be the set of edges of  $T'$  (see Figure 3-a). Given any valid budget allocation  $\mathcal{B}$  to  $E$ , let  $\mathcal{B}'$  be the scaling of the restriction of  $\mathcal{B}$  to  $E'$ , defined by  $\mathcal{B}'(e') = \frac{\mathcal{B}(e')}{1-\mathcal{B}(e)}$  for every  $e' \in E'$ . Note that with this scaling,  $\mathcal{B}'$  is a valid budget allocation to  $E'$ , i.e.,  $\sum_{e' \in E'} \mathcal{B}'(e') = 1$ . Since any path from  $r$  to any leaf of  $T$  must start with the edge  $e = (r, r')$ , it follows that

$$\text{wr}_{\mathcal{B}}(r) = \frac{q}{\mathcal{B}(e)} + \frac{\text{wr}_{\mathcal{B}'}(r')}{1 - \mathcal{B}(e)}.$$

Hence, for  $\mathcal{B}$  to be optimal for  $T$  with root  $r$ , we must have  $\mathcal{B}'$  be optimal for  $T'$  and  $r'$ . In addition, given  $R' = \text{BR}(T', r')$ , the budget radius of  $T$  with root  $r$  is obtained by assigning a  $\beta$  fraction of the budget to  $e$ , for  $\beta$  that minimizes the function  $\text{wr}_{\mathcal{B}}(r) = \frac{q}{\beta} + \frac{R'}{1-\beta}$ . It follows that  $\text{BR}(T, r) = \frac{q}{\beta} + \frac{R'}{1-\beta}$  for  $\beta = \frac{\sqrt{q}}{\sqrt{R'} + \sqrt{q}}$ .  $\square$

We next consider the general family of (rooted) trees (Figure 3-b). Let  $T = (V, E)$  be a tree, rooted at  $r$ , such that  $r$  has  $k$  children  $r_1, r_2, \dots, r_k$ , where  $r_i$  is the root of the subtree  $T_i = (V_i, E_i)$ . Denote by  $T'_i = (V'_i, E'_i)$  the subtree of  $T$ , rooted at  $r$  and containing  $T_i$ . Formally,  $V'_i = V_i \cup \{r\}$ , and  $E'_i = E_i \cup \{(r, r_i)\}$ . Clearly, the edges sets  $E'_i$ s are disjoint. Given a valid budget allocation  $\mathcal{B}$  to  $E$ , for each index  $i \in \{1, 2, \dots, k\}$  denote by  $l_i \in V_i$  the leaf  $l$  for which  $\delta_{\mathcal{B}}(r, l)$  is the largest within  $T'_i$ . Recall that the weighted radius  $\text{wr}_{\mathcal{B}}(r)$  is determined by the maximum weighted distance to some  $l_i$ . In other words,  $\text{wr}_{\mathcal{B}}(r) = \max_{1 \leq i \leq k} (\delta_{\mathcal{B}}(r, l_i))$ . Next, we show that in any optimal budget allocation  $\mathcal{B}^*$  for such  $T$ , the fraction of the budget assigned to the edges of each subtree  $T'_i$  is directly correlated to its relative weighted radius.



**Figure 3:** (a) The case that  $r$  has only a single child. (b) The general case.

**Lemma 3.3.** *Let  $T = (V, E)$  be a tree rooted at  $r$ , with some length function  $\ell$  on the edges of  $T$ . Assume that  $r$  has  $k$  children  $r_1, r_2, \dots, r_k$  where  $r_i$  is the root of the subtree  $T_i = (V_i, E_i)$ , and let  $\mathcal{B}^*$  be an optimal budget allocation to  $E$ . For each  $1 \leq i \leq k$ , let  $T'_i$  and  $l_i$  be as in the foregoing discussion (i.e.,  $l_i$  is maximal in  $T'_i$  with respect to  $\delta_{\mathcal{B}^*}(r, \cdot)$ ). Then, for all  $1 \leq i, j \leq k$  that  $\delta_{\mathcal{B}^*}(r, l_i) = \delta_{\mathcal{B}^*}(r, l_j)$ .*

*Proof.* Assume that for some  $1 \leq i, j \leq k$ , it holds that  $\delta_{\mathcal{B}^*}(r, l_i) > \delta_{\mathcal{B}^*}(r, l_j)$ . We show that it is then possible to present a better budget allocation for  $T$ , which, in turn, leads to a contradiction. Let  $\rho = \frac{\delta_{\mathcal{B}^*}(r, l_j)}{\delta_{\mathcal{B}^*}(r, l_i)}$  and consider an alternative budget allocation in which each edge  $e$  in  $E'_j$  was assigned a  $\rho$  fraction of its current budget, i.e.  $\rho \cdot \mathcal{B}^*(e)$  (while assignment to all other edges stays the same as before). The length of each path from  $r$  to a leaf in  $T'_j$  would be multiplied by  $1/\rho$ . Hence, the maximum distance from  $r$  to any leaf in the  $T'_j$  would be at most  $\delta_{\mathcal{B}^*}(r, l_i)$ . This allocation is therefore as good as  $\mathcal{B}^*$  (with respect to the weighted radius) although the sum of assigned values is not 1, but rather,  $1 - (1 - \rho) \cdot \mathcal{B}^*(E'_j) < 1$ . Turning it into a valid budget allocation by dividing the remaining  $(1 - \rho) \cdot \mathcal{B}^*(E'_j)$  budget equally among all edges in  $E$ , we obtain a better valid budget allocation to  $E$ . That is, we fix a new allocation  $\mathcal{B}'$  by setting  $\mathcal{B}'(e) = \rho \cdot \mathcal{B}^*(e) + \frac{(1 - \rho) \cdot \mathcal{B}^*(E'_j)}{|E|}$  if  $e \in E'_j$ , and  $\mathcal{B}'(e) = \mathcal{B}^*(e) + \frac{(1 - \rho) \cdot \mathcal{B}^*(E'_j)}{|E|}$  otherwise. The budget radius of  $\mathcal{B}'$  is smaller than that of  $\mathcal{B}^*$ , in contradiction to the optimality of  $\mathcal{B}^*$ .  $\square$

The following corollary describes how any optimal valid budget allocation must divide the budget among the disjoint sets of edges of the subtrees  $T'_i$ .

**Corollary 3.4.** *Let  $T$  be a tree as above. Then in any optimal budget allocation  $\mathcal{B}^*$  to  $E$  it holds that  $\mathcal{B}^*(E'_i) = \frac{\text{BR}(T'_i, r)}{\sum_{j=1}^k \text{BR}(T'_j, r)}$ . Thus, an optimal solution in this case is given by  $\text{BR}(T, r) = \sum_{j=1}^k \text{BR}(T'_j, r)$ .*

*Proof.* Let  $l_i$ 's be as above, i.e.,  $l_i \in V_i$  is the leaf  $l$  for which  $\delta_{\mathcal{B}^*}(r, l)$  is maximal within  $T'_i$ . Denote  $\beta_i = \mathcal{B}^*(E'_i)$  and let  $\mathcal{B}'_i$  be the scaling of the restriction of  $\mathcal{B}^*$  to  $E'_i$ , defined by  $b'_i(e') = \frac{\mathcal{B}^*(e')}{\beta_i}$  for every  $e' \in E'_i$ . Clearly, each  $\mathcal{B}'_i$  is a valid budget allocation to  $T'_i$ . We claim that it is also an optimal one. By Lemma 3.3, for all  $1 \leq i, j \leq k$  it holds that  $\delta_{\mathcal{B}^*}(r, l_i) = \delta_{\mathcal{B}^*}(r, l_j)$ . If for some  $i$  it holds that  $\mathcal{B}'_i$  is not optimal for  $T'_i$ , then choose an optimal valid budget allocation  $\mathcal{B}''_i$  for  $T'_i$  and scale it back to obtain a valid budget allocation  $\hat{\mathcal{B}}$  by setting  $\hat{\mathcal{B}}(e') = \mathcal{B}''_i(e') \cdot \beta_i$  for each  $e' \in E'_i$ , and  $\hat{\mathcal{B}}(e) = \mathcal{B}^*(e)$  for each  $e' \notin E'_i$ . This reduces the distance of the farthest leaf from  $r$  within  $T'_i$ , while the distance to any leaf outside  $T'_i$  stays as with  $\mathcal{B}^*$ . Specifically, we have  $\text{wr}_{\hat{\mathcal{B}}}(r) \leq \text{wr}_{\mathcal{B}^*}(r)$ . However, by Lemma 3.3,  $\hat{\mathcal{B}}$  is not optimal and hence,  $\mathcal{B}^*$  is not optimal either – contradiction.

By the above it holds for all  $1 \leq i \leq k$  that  $\text{BR}(T'_i, r) = \beta_i \cdot \delta_{\mathcal{B}^*}(r, l_i)$ . Thus, by Lemma 3.3, for all  $1 \leq i, j \leq k$  it holds that  $\frac{\text{BR}(T'_i, r)}{\beta_i} = \frac{\text{BR}(T'_j, r)}{\beta_j}$ . Since it also holds that  $\sum_{i=1}^k \beta_i = 1$ , we have that for all  $1 \leq i \leq k$  it holds that  $\beta_i = 1 - \sum_{j \neq i} \beta_j = 1 - \beta_i \cdot \sum_{j \neq i} \frac{\text{BR}(T'_j, r)}{\text{BR}(T'_i, r)}$ . Hence,  $\beta_i = \frac{\text{BR}(T'_i, r)}{\sum_{j=1}^k \text{BR}(T'_j, r)}$ . Furthermore, since for any  $1 \leq i \leq k$  we have that  $\text{BR}(T, r) = \delta_{\mathcal{B}^*}(r, l_i)$  (specifically, since  $\text{BR}(T, r) = \delta_{\mathcal{B}^*}(r, l_1) = \frac{\text{BR}(T'_1, r)}{\beta_1}$ ), it follows that  $\text{BR}(T, r) = \sum_{j=1}^k \text{BR}(T'_j, r)$ .  $\square$

**Theorem 3.5.** *Given a tree  $T$  rooted at  $r$ , it is possible to find an optimal valid budget allocation for  $T$  and  $r$ , in linear time in the size of  $T$ .*



*Proof.*  $T$  is a rooted tree, thus an inductive construction is only natural. First, assume  $T$  is a single node  $r$ . In this case, no budget is needed and  $\text{BR}(T, r) = 0$ . Assume  $T$  is rooted at  $r$ , such that  $r$  has  $k$  children  $r_1, r_2, \dots, r_k$ . Denote by  $T_i$  the subtree of  $T$  rooted at  $r_i$ , and containing all vertices (and edges) of the subtree rooted at  $r_i$  (and only these vertices). Denote by  $T'_i = (V'_i, E'_i)$  the subtree of  $T$  rooted at  $r$ , induced by adding the edge  $(r, r_i)$  to  $T_i$ . Formally,  $V'_i = V_i \cup \{r\}$ , and  $E'_i = E_i \cup \{(r, r_i)\}$ . Thus, each  $T'_i$  is a rooted tree where the root ( $r$ ) has a single child, and no  $T'_i, T'_j$  for  $i \neq j$  share any vertex other than  $r$  and  $E'_i, E'_j$  are disjoint for all  $i \neq j$ .

By , if we know  $\text{BR}(T'_i, r)$  for all  $1 \leq i \leq k$ , we can derive an optimal valid budget allocation for  $T, r$ . In order to obtain a  $\text{BR}(T'_i, r)$ , it suffices to have an optimal solution for the subtree of  $r_i$ , which, by the induction hypothesis can be done (using Lemma 3.2).

Note, that we evaluate the optimal solution for every subtree of every vertex in  $T$  exactly once and thus the procedure requires in linear time.  $\square$

The following lemma proves helpful in the sequel, but is interesting in its own right. It captures some of the tricky nature of the budget radius problem, as it shows the connection between two seemingly unrelated quantities. The first is the weight of a minimum spanning tree (MST) of a given graph and the second is the optimal solution for the budget radius problem for that graph.

**Lemma 3.6.** *Given a tree  $T = (V, E)$  rooted at  $r$ , with some length function  $\ell$  on  $E$ , the budget radius of  $T$  is at least the sum of lengths of the edges of  $T$ , i.e.,  $\text{BR}(T, r) \geq \sum_{e \in E} \ell(e)$ .*

*Proof.* We prove the above lemma by induction. If  $T$  has no edges, then both values are 0. If  $r$  has only one child  $r'$  (the root of the subtree  $T'$ ), then by Lemma 3.2, since any optimal allocation  $\mathcal{B}^*$  must assign  $\mathcal{B}^*((r, r')) > 0$ , we have  $\text{BR}(T, r) > \ell((r, r')) + \text{BR}(T', r')$ , which, by the induction hypothesis is at least  $\sum_{e \in E} \ell(e)$ . Otherwise, assume  $r$  has  $k$  children ( $r_1 \dots r_k$ ) and denote  $T_i$  the subtree induced by  $r$  and the vertices of the subtree of  $r_i$ . By  $\text{BR}(T, r) = \sum_{i=1}^k \text{BR}(T_i, r)$ . Hence, by the induction hypothesis, the lemma follows.  $\square$

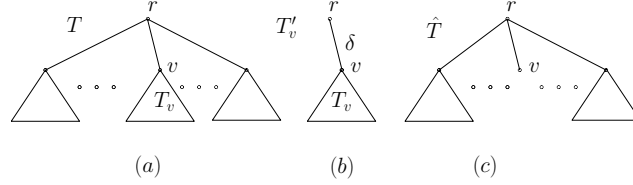
### 3.2. The Budget Radius for Unrooted Trees

In this section we consider the budget radius problem for unrooted trees, i.e., where the root of the tree is not given as part of the input. Clearly, one can invoke the algorithm from Theorem 3.5 with every vertex  $v$  as a candidate center vertex  $r$ , and select the vertex  $v$  for which  $\text{BR}(T, v)$  is minimal as the ultimate center. However, this naive algorithm requires  $O(n^2)$  time. We next show how to construct a linear time algorithm for this problem (indeed, for a tree  $T$ , our algorithm computes  $\text{BR}(T, v)$  for every  $v$  in  $T$ ). Intuitively, this protocol uses the fact that given  $\text{BR}(T, r)$  and the partial computations made by algorithm of Theorem 3.5, applied to the  $T$  and  $r$ , it possible to compute in constant time  $\text{BR}(T, v)$  for every neighbor  $v$  of  $r$ . This intuition is formalized in Lemma 3.7.

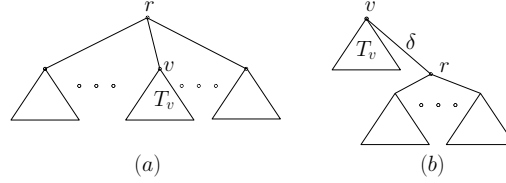
**Lemma 3.7.** *Let  $T = (V, E)$  be a tree rooted at  $r$ , with some length function  $\ell$  on  $E$ . Let  $v \in V$  be a neighbor (a child) of  $r$ . Denote by  $T_v = (V_v, E_v)$  the subtree of  $v$ , and denote by  $T'_v$  the subtree of  $v$  augmented by the edge  $e = (r, v)$  (i.e.,  $T'_v = (V_v \cup r, E_v \cup e)$ ). It is possible to compute, in constant time,  $\text{BR}(T, v)$  given  $\text{BR}(T, r)$ ,  $\text{BR}(T_v, v)$ , and  $\text{BR}(T'_v, r)$ , see Figure 4.*

*Proof.* Denote by  $\hat{T}$  the tree obtained by omitting  $T_v$  from  $T$ , formally,  $\hat{T} = (\hat{V}, \hat{E})$ , where  $\hat{V} = V \setminus (V_v \setminus \{v\})$  and  $\hat{E} = E \setminus E_v$ . In addition, denote by  $\hat{T}'$  the tree obtained by omitting the edge  $e = (r, v)$  from  $\hat{T}$ , i.e.,  $\hat{T}' = (\hat{V} \setminus \{v\}, \hat{E} \setminus \{e\})$ .

It can be easily derived from that  $\text{BR}(T, v) = \text{BR}(T_v, v) + \text{BR}(\hat{T}, v)$ . By Lemma 3.2, we can compute  $\text{BR}(\hat{T}, v)$  from  $\text{BR}(\hat{T}', r)$  and  $\ell(e)$ , in constant time. Finally, we compute  $\text{BR}(\hat{T}', v)$ , using again, to obtain  $\text{BR}(\hat{T}', r) = \text{BR}(T, r) - \text{BR}(T_v, r)$ .  $\square$



**Figure 4:** (a) The original tree rooted at  $r$ . (b) Considering  $v$  as the root of  $T'_v$ . (c) The tree  $\hat{T}$ .



**Figure 5:** (a) The original tree rooted at  $r$ . (b) Considering  $v$  as the root of the tree. Computation of an optimal budget allocation for the tree rooted at  $v$  can be done in constant time, given an optimal budget allocation for the tree rooted at  $r$ .

Roughly, our algorithm will traverse the tree twice. First, we traverse the tree, computing the algorithm of Theorem 3.5 for an arbitrary root  $r$  (say,  $r = v_1$ ). Recall that this algorithm traverses the tree in a bottom-up fashion, i.e., from the leaves to the root, and that an optimal solution for each vertex is calculated, with respect to the subtree below it. Thereafter, we traverse the tree in a top-down fashion, while for each vertex  $v$  that is a child of  $v'$ , we compute an optimal budget radius for the tree with root  $v$ , given an optimal budget radius for the tree with root  $v'$  and the information stored in  $v$  from the first traversal.

**Theorem 3.8.** *Given a tree  $T = (V, E)$  with some length function  $\ell$  on  $E$ , it is possible to compute an optimal allocation for  $T$ , i.e., a pair  $(\mathcal{B}^*, r^*)$ , such that  $\text{wr}_{\mathcal{B}^*}(r^*) = \text{BR}(T)$ . Furthermore, this can be done in linear time in the size of  $T$ .*

*Proof.* Our algorithm traverses the tree twice. In the first pass, we set an arbitrary vertex  $r$  to be the root (say,  $r = v_1$ ) and traverse the tree in a bottom up manner, following the algorithm described in Theorem 3.5.

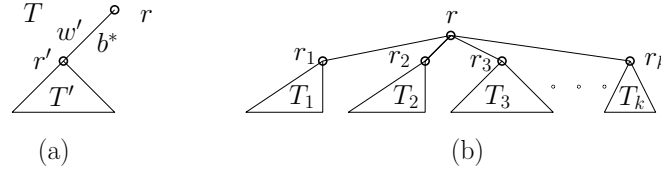
For any vertex  $v \in V$ , denote by  $T_v = (V_v, E_v)$  the subtree of  $v$ , and denote by  $p(v)$  the parent of  $v$ . Denote by  $T'_v$  the subtree of  $v$  augmented by the edge  $(p(v), v)$ , i.e.,  $T'_v = (V_v \cup \{p(v)\}, E_v \cup \{(p(v), v)\})$ . We compute for each vertex  $v$  the value of an optimal budget radius with respect to the subtree of  $v$ , i.e.,  $\text{BR}(T_v, v)$ . Recall that in order to do so, we compute for each child  $u$  of  $v$ , not only the budget radius for  $T_u$  (i.e.,  $\text{BR}(T_u, u)$ ), but also the budget radius for the augmented subtree of  $u$  (with root  $v$ ), i.e.,

$\text{BR}(T'_u, v)$ . Here, we also store the two local values,  $\text{BR}(T_v, v)$  and  $\text{BR}(T_{p(v)}, p(v))$ , for each node  $v$  we traverse.

In the second pass we traverse the tree in a top-down manner, starting at the root  $r$  and moving from each vertex to all its children. For each vertex  $v$ , we compute the budget radius for the whole tree  $T$  with root  $v$  (i.e.,  $\text{BR}(T, v)$ ). By Lemma 3.7, we can do so in constant time since we have already computed  $\text{BR}(T, p(v))$ , as well as  $\text{BR}(T_v, v)$  and  $\text{BR}(T_{p(v)}, p(v))$ . The algorithm returns the pair  $(\mathcal{B}^*, r^*)$ , for which, the budget radius is minimal.  $\square$

#### 4. Generalization to Median Point

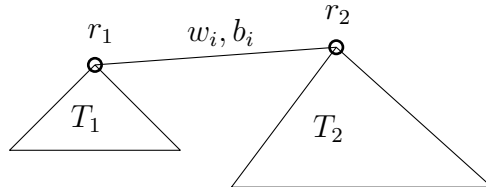
In this section we generalize the center point algorithms to the *median point*. In this case we would like to find the node ( $M$ ) and its corresponding budget allocation which minimizes the average (or sum) weight of all shortest path from  $M$  to all the graph nodes. Following the same framework as in the center point we now would like to find the optimal allocation for  $\mathcal{B}^*$ , for a fixed node  $r$  as a median point.



**Figure 6:** Following the same framework as in the center point we now would like to find the optimal allocation for  $\mathcal{B}^*$ . (a) Single subtree case:  $b^*$  minimizes the value of:  $\frac{\text{sum}(T', r')}{1-b} + n \frac{w'}{b}$ , where  $n = |T'|$ . (b) Multiple subtrees case: can be solved as an  $LP$  minimization problem.

In case there is a single child to the root  $r$  (see Figure 6a) we need to minimize the following term:  $\text{sum}(T, r) = \frac{\text{sum}(T', r')}{1-b^*} + n \frac{w'}{b^*}$ , where  $n = |T'|$ . Observe that  $\text{sum}(T, r) = \frac{c_1 + \frac{c_2}{x}}{1-x}$  where  $c_1 = \text{sum}(T', r')$ ,  $c_2 = nw'$  therefore  $\mathcal{B}^*$  can be computed in constant time, and so is  $\text{sum}(T, r)$ .

If the root  $r$  has more than a single subtree (as shown in Figure 6b), we can apply the above computation on each subtree independently, calculating the optimal budget allocation for each edge  $e_i = (r_i, r)$ , then the budget between all subtrees can be normalized using the following minimization problem:  $\min(\frac{X_1}{B_1} + \frac{X_2}{B_2} + \dots + \frac{X_k}{B_k})$  where  $B_i > 0$ ,  $X_i$  is the optimal sum of  $T_i$  with the root  $r$ , and  $B_1 + B_2 + \dots + B_k = 1$ . This problem can be solved in  $O(k)$  time.



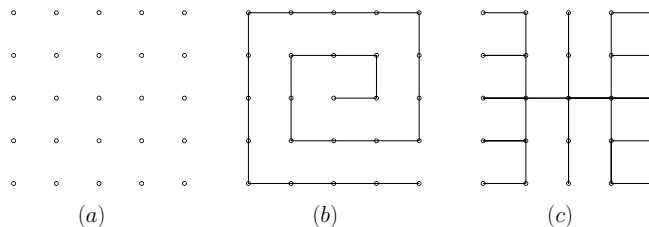
**Figure 7:** Assuming  $|T_1| < |T_2|$  the median cannot be in  $T_1$ , regardless of  $w_i$  and  $b_i$ .

**Corollary 4.1.** *The location of the median in the unrooted tree case depends only on the tree structure, i.e., it is in the regular unweighted tree median node. This is due to the*

convex nature of the median see figure 7. Therefore, the optimal root for the median point can be found in linear time.

## 5. Budget Radius – The General Case

In this section we consider the general case problem of optimizing the budget radius for a complete graph over  $n$  vertices, induced by some metric space  $M = (V, d)$ . We present a hardness proof showing that the general case of the budget radius problem is N.P. hard. We then present an  $O(\log^2(n))$  approximation algorithm for this problem. In order to motivate the non-triviality of a logarithmic approximation factor, We start by showing that a naive Minimum Spanning Tree (MST) heuristic may lead to an  $O(n^{0.5})$  approximation factor. Assume we have  $n$  points on a square uniform grid. Its *MST* may have a path like shape, with  $\Omega(n)$  radius. Hence its budget radius is  $\Omega(n^2)$ . On the other hand, each of the  $n$  points may be connected to the center with a path of length  $O(n^{0.5})$ . Hence, the budget radius of this metric is  $O(n^{1.5})$ .



**Figure 8:** Using an MST-like heuristic may lead to an  $O(n^{0.5})$  approximation ratio with respect to the *center point* problem (minimum radius). (a) A grid based set of points. (b) A path-like MST. (c) A solution with radius  $O(n^{0.5})$ .

### 5.1. Hardness results

In this section we present a reduction showing that finding the optimal budget allocation for radius budget problem on rooted graphs is N.P. hard problem. Let  $G = (V, E)$  be an undirected graph induced by some length function  $\ell(e)$  for each  $e \in E$ . Let  $B$  be a positive budget value. Allocating a budget  $B(e)$  to edge  $e \in E$  with length  $\ell(e)$  implies that the resulting weight of  $e$  is  $w(e) = \frac{\ell(e)}{B(e)}$ . Let  $(G, d_w)$  denote graph  $G$  equipped with a distance function  $d_w(\cdot, \cdot)$ , defined as the minimum path distance under weight function  $w(\cdot)$ . The rooted budget radius problem is defined as follows: Given vertex  $r \in V$ , allocate budget  $B$  among the edges of  $E$  (i.e.,  $\sum_{e \in E} B(e) = B$ ) so that the set radius  $\max_{v \in V} d_w(r, v)$  is minimized. We prove the following theorem:

**Theorem 5.1.** *The decision version of the rooted budget radius problem is NP-hard.*

Before proving Theorem 5.1, we require a preliminary lemma and technical observation:

**Lemma 5.2.** *Let  $G = (V, E)$  be a star graph with  $V = \{r, s, v_1, \dots, v_k\}$ . Let  $s$  be the center of the star, and let edge  $e = (r, s)$  have length  $\ell(e) = 1$  and the  $k$  edges  $e_i = (s, v_i)$  (for  $i = 1, \dots, k$ ) have length  $\ell(e_i) = x$ . Then in the optimal solution to the rooted budget radius problem on  $G$  the radius equals*

$$\frac{(1 + \sqrt{xk})^2}{B}$$

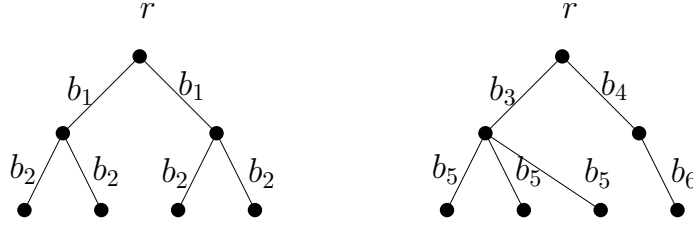
*Proof.* First note that in the optimal solution, the path from  $r$  to each  $v_i$  is equal to  $w(e) + w(e_i)$ . Hence, it must be that all  $k$  allocations  $B(e_i)$  are equal, as otherwise not all  $w(e_i)$  are equal, and then some weight could be taken from the largest allocation  $B(e_j)$  (for some  $j \in [1, k]$ ) and distributed evenly among all allocations  $B(e_i)$ , resulting in a shorter radius.

Set  $y = B(e)$ , and it must be that  $B(e_i) = \frac{B-y}{k}$ . It follows that the optimal solution minimizes

$$w(e) + w(e_i) = \frac{1}{y} + \frac{x}{(B-y)/k}.$$

This term is minimized when  $\frac{1}{y^2} = \frac{xk}{(B-y)^2}$ , that is  $y = \frac{B}{1+\sqrt{xk}}$ . So  $B(e) = y = \frac{B}{1+\sqrt{xk}}$  and  $w(e) = \frac{1}{b(e)} = \frac{1+\sqrt{xk}}{B}$ , while  $B(e_i) = \frac{B-y}{k} = \frac{B}{k} \cdot \frac{\sqrt{xk}}{1+\sqrt{xk}}$  and  $w(e_i) = \frac{x}{b(e_i)} = \frac{\sqrt{xk}(1+\sqrt{xk})}{B}$ . The radius is  $w(e) + w(e_i) = \frac{(1+\sqrt{xk})^2}{B}$ .  $\square$

**Observation 5.3.** Consider Figure 9, The optimal allocations for the left tree (a) and the right tree (b) are not the same, the right tree has a smaller budget radius. In fact, the optimal allocation of tree (a) can be used on the right tree (b) leading to the same radius (allocating to  $b_3 = b_4 = b_1$  and  $b_5 = b_6 = b_2$ ). The optimal budget allocation for both trees is presented in figure 1 cases: D, E).



**Figure 9:** The radius budget of the right tree (b) is smaller than the left tree (a).

We now return to the proof of Theorem 5.1:

*Proof.* The theorem is proved via reduction to 3-Set Cover, the version of Set Cover where the cardinality of each set is at most 3. 3-Set Cover is NP-hard. In the reduction, we will assume that in the minimum radius problem the target radius is fixed and the goal is to find the minimum cost  $B$  – by scaling, this is equivalent to the scenario where  $B$  is fixed as the goal is to minimize the radius.

Given a set cover instance  $(S, E)$  of sets and elements, we create a graph  $G$  as follows. First create root node  $n_r$ . For each element  $e_i$  in the set cover instance, create a single node  $n_{e_i}$ . We call these nodes *element-nodes*. For each 3-Set  $S_j = \{e_a, e_b, e_c\}$ , we create seven nodes called *set-nodes*. Set  $x = 1$ ,  $y = \frac{(\sqrt{6}-1)^2}{2}$  and  $z = \frac{(\sqrt{8}-1)^2}{3}$ :

1. A new set-node  $n_{S_j, e_a}$  is connected to root  $n_r$  by an edge of length 1, and to element-node  $n_{e_a}$  by an edge of length  $x$ . New set-nodes  $n_{S_j, e_b}$  and  $n_{S_j, e_c}$  are similarly connected to  $n_r$ , and respectively to element-nodes  $n_{e_b}$  and  $n_{e_c}$ . These set-nodes will represent the solution case where set  $S_j$  is needed to cover only a single element.
2. A new set-node  $n_{S_j, e_a, e_b}$  is connected to root  $n_r$  by an edge of length 1, and to element-nodes  $n_{e_a}$  and  $n_{e_b}$  by edges of length  $y$ . New set-nodes  $n_{S_j, e_a, e_c}$  and  $n_{S_j, e_b, e_c}$  are similarly connected to  $n_r$ , and respectively to element-node pairs  $n_{e_a}$  and  $n_{e_c}$  or

$n_{e_b}$  and  $n_{e_c}$ . These set-nodes will represent the solution case where set  $S_j$  is needed to cover only two elements.

3. A new set-node  $n_{S_j, e_a, e_b, e_c}$  is connected to root  $n_r$  by an edge of length 1, and to each of element-nodes  $n_{e_a}, n_{e_b}, n_{e_c}$  by an edge of length  $z$ . This set-node represents the solution case where set  $S_j$  is needed to cover all three elements.

Let the set of set-nodes for  $S_j$  be  $N_j$ .

Let the radius in the optimal solution be 1, and we will show that finding the minimum cost  $B$  achieving this radius on  $G$  is equivalent to solving set cover on the input instance.

In the optimal solution, for each element-node  $n_{e_a}$  only one edge incident on  $n_{e_a}$  is assigned non-zero weight: At least one edge must be assigned weight, or else the path from the root  $n_r$  to  $n_{e_a}$  is infinite. Further, if the optimal solution assigns weight to two edges incident on  $n_{e_a}$ , then it must be that one of the incident set-nodes has its minimum path to  $n_r$  routed through  $n_{e_a}$ . But this forms a contradiction: The edge connecting this set-node to  $n_r$  has length  $1 = x < y < z$ , so a shorter path can be attained by removing the weight from the edge connecting  $n_{e_a}$  to the set-node, and placing it on the edge connecting the set-node to  $n_r$ .

It follows that in the optimal solution, the only finite paths are those consisting of a single edge connecting the root to a set-node, and the edges connecting the set-node to at most three element-nodes. So the the set of finite edges form a set of star graphs connected at  $n_r$ .

Now consider a set  $N_j$  of set-nodes for set  $S_j$ , and all edges incident on this set. Suppose there are exactly  $c$  element-nodes whose minimum paths are routed through set-nodes of  $N_j$ . Clearly  $0 \leq c \leq 3$ .

1. If  $c = 0$ , then weight is assigned only to the length 1 edges connecting the seven set-nodes of  $N_j$  directly to the root, at a total cost of 7.
2. If  $c = 1$ , then since  $x < y < z$ , weight is assigned to the length  $x$  edge connecting the relevant element-node  $n_{e_a}$  to set-node  $n_{S_j, e_a}$ . Consider the star graph formed by nodes  $n_r, n_{S_j, e_a}, n_{e_a}$ : By Lemma 5.2, to achieve a path length of 1 the total weight assigned to the two graph edges is  $(1 + \sqrt{x})^2 = 4$ . The other six edges connecting  $n_r$  to  $N_j$  are each assigned weight 1, for a total cost of  $6 + (1 + \sqrt{x})^2 = 10$ .
3. If  $c = 2$ , the optimal solution will assign weight to the length  $y$  edges connecting the relevant element-nodes  $n_{e_a}, n_{e_b}$  to set-node  $n_{S_j, e_a, e_b}$ . Consider the star graph formed by nodes  $n_r, n_{S_j, e_a, e_b}, n_{e_a}, n_{e_b}$ : By Lemma 5.2, to achieve path lengths of 1 the total weight assigned to the three graph edges is  $(1 + \sqrt{2y})^2 = 6$ . The other six edges connecting  $n_r$  to  $N_j$  are each assigned weight 1, for a total cost of  $6 + (1 + \sqrt{2y})^2 = 12$ .

Another option would have been to assign weight to the two edges connecting  $n_{e_a}$  to  $n_{S_j, e_a}$  and  $n_{e_b}$  to  $n_{S_j, e_b}$ . In this case, the total cost is  $5 + 2(1 + \sqrt{x})^2 = 13 > 12$ , so this assignment is suboptimal.

4. If  $c = 3$ , the optimal solution will assign weight to the length  $z$  edges connecting element-nodes  $n_{e_a}, n_{e_b}, n_{e_c}$  to set-node  $n_{S_j, e_a, e_b, e_c}$ . Consider the star graph formed by nodes  $n_r, n_{S_j, e_a, e_b, e_c}, n_{e_a}, n_{e_b}, n_{e_c}$ : By Lemma 5.2, to achieve path lengths of 1 the total weight assigned to the three graph edges is  $(1 + \sqrt{3z})^2 = 8$ . The other six edges connecting  $n_r$  to  $N_j$  are each assigned weight 1, for a cost of  $6 + (1 + \sqrt{3z})^2 = 14$ .

Another option would have been to assign weight to three edges connecting  $n_{e_a}$  to  $n_{S_j, e_a}$ ,  $n_{e_b}$  to  $n_{S_j, e_b}$ , and  $n_{e_c}$  to  $n_{S_j, e_c}$ . In this case, the total cost is  $4 + 3(1 + \sqrt{x})^2 = 16 > 14$ , so this assignment is suboptimal. If we would assign weight to the edges connecting  $n_{e_a}$  to  $n_{S_j, e_a}$  and both  $n_{e_b}, n_{e_c}$  to set-node  $n_{S_j, e_b, e_c}$ , then the total weight would be  $5 + (1 + \sqrt{x})^2 + (1 + \sqrt{2y})^2 = 15 > 14$ , so this assignment is suboptimal.

Recall that in the optimal solution, each element-node is incident on a single edge with non-zero weight, and this edge connects the element-node to a set-node. We will say that the set-node *covers* this element node. As above, for a given set  $N_j$ , at most one set-node in  $N_j$  covers some element-nodes. In this case we say that  $N_j$  is *used*, and that  $N_j$  covers these element-nodes.

We may then view the above cost assignment as follows: For each set  $N_j$ , if the  $N_j$  is not used we pay a single overhead cost of 7, and if  $N_j$  is used we pay a overhead cost of 8 plus a cost of 2 for each element-node covered by  $N_j$ . Then the total cost for the graph is  $7|S| + 2|E|$  plus the number of used sets, and the minimum cost is achieved by using the minimum number of set-nodes to cover all element-nodes. This is equivalent to solving the Set Cover problem on the input.  $\square$

## 5.2. Approximation Algorithm - General Case of Budget Radius

### 5.2.1. The Special Case of a Line

We first consider a setup in which  $M$  is defined by some  $n$  points all residing on the interval  $[0, 1]$ , where for any two points  $p_1, p_2$  within this interval,  $d(p_1, p_2)$  is the Euclidean distance between  $p_1$  and  $p_2$ . Let  $G = (V, E)$  be the complete graph induced by  $M$ . We present a valid budget allocation  $\mathcal{B}$  to  $E$  with budget radius at most  $\log^2 n$  and such that the graph induced by  $\{e : \mathcal{B}(e) > 0\}$  is a tree spanning  $V$ .

**Lemma 5.4.** *Let  $G = (V, E)$  be the complete graph described above, then  $\text{BR}(G) \leq \log^2 n$ .*

*Proof.* Let  $P = \{p_1, p_2, \dots, p_n\}$  be a set of  $n$  points on the interval  $[0, 1]$  (in increasing order). Next, we construct a full binary tree  $T$  over  $P$ . Its root is  $p_{\frac{n}{2}}$ , and the root's children are  $p_{\frac{n}{4}}$  and  $p_{\frac{3n}{4}}$ , etc.

The set of possible solutions for the budget radius problem for  $T$  is a subset of the set of possible solutions for the budget radius problem for  $G$ . Thus, it suffices to present a solution for  $T$  with budget radius at most  $\log^2(n)$ , that is, a pair of the form  $(\mathcal{B}, r')$ , where  $\mathcal{B}$  is a valid budget allocation to  $E_T$  and  $r' \in V_t$ , such that,  $\text{wr}_{\mathcal{B}}(r') \leq \log^2(n)$ . Clearly, fixing  $r' = r$  only further restricts the set of solutions we allow. We next describe one such solution.

We first divide the edges of  $T$  into sets, defined by the depth of a given edge from the root  $r$ . Formally, if  $e$  is an edge in  $T$ , we say that  $e$  is at level  $i$  in  $T$  if one of its vertices has depth  $i$  and the other has depth  $i + 1$ . We denote the set of all edges in  $T$  of level  $i$  by  $S_i$ . For every edge  $e \in S_i$  we set  $\mathcal{B}(e) = \alpha_e$ , where  $\alpha_e \stackrel{\text{def}}{=} \frac{1}{\log(n)} \frac{\ell(e)}{\sum_{e' \in S_i} \ell(e')}$ . We first need to show that this allocation is valid and sums up to at most 1. This is true since for every level  $i$ , we divide a  $\frac{1}{\log(n)}$  fraction of the budget among the edges in  $S_i$ . Since there are no more than  $\log(n)$  levels, we do not exceed our budget.

To bound the budget radius of  $T$  under this allocation, observe that as  $T$  is a search tree, it holds for every  $i$  that  $\sum_{e' \in S_i} \ell(e') \leq 1$ . Thus, for every edge  $e$  of  $T$ , we have  $\alpha_e \geq \frac{\ell(e)}{\log(n)}$ . Now, let  $P$  be a simple path from  $r$  to some leaf  $\ell$ . The weighted length of

$P$  (the weighted distance between  $r$  and  $\ell$ ) is  $\sum_{e \in P} \omega_{\mathcal{B}}(e) = \sum_{e \in P} \frac{\ell(e)}{\alpha_e} \leq \sum_{e \in P} \log(n)$ , which is at most  $\log^2(n)$  since  $P$  consists of at most  $\log(n)$  edges.  $\square$

### 5.2.2. General Complete (Metric) Graphs

We next define an approximation algorithm  $\mathcal{A}$ , such that given a complete graph  $G = (V, E)$ , induced by some metric space  $M = (V, d)$ , approximates the Budget Radius problem for  $G$  by a factor of  $O(\log^2 n)$ . Assume that a minimum spanning tree for  $G$  has a total weight LB, we proceed as follows:

1. Find an Hamiltonian path ( $HP$ ) visiting all nodes with weight no more than  $2 \cdot \text{LB}$ .
2. Let  $G'$  be the result of unfolding  $HP$  to a straight line, i.e.,  $G'$  is defined by  $n$  points, situated on an interval, such that, the distance between every two points is the length of the path between them on the Hamiltonian path  $HP$  (specifically, the length of the whole interval is exactly the length of  $HP$ ).
3. Scale the above ( $HP$ ) interval length to 1.
4. Build a balanced binary search tree (BT) over  $G'$ .
5. Apply the algorithm of Theorem 3.5 to BT. Assign the appropriate budget to all edges in BT and 0 to all other edges in  $E$ .

**Theorem 5.5.** *Let  $G = (V, E)$  be a complete graph induced by some metric space  $M = (V, d)$ . Then, algorithm  $\mathcal{A}$  results in a valid budget allocation to the edges of  $E$  that approximates  $\text{BR}(G)$  by a  $2 \log^2(n)$  factor.*

*Proof.* First note that finding an Hamiltonian path ( $HP$ ) with weight no more than  $2 \cdot \text{LB}$  is feasible using an MST for  $G$ . More importantly, note that by Lemma 3.6, it holds that LB is a lower bound on the optimal solution (i.e., on  $\text{BR}(G)$ ). This is true since an optimal budget allocation defines a tree (see Lemma 3.1), which has a total weight of at least LB (by the minimality of an MST). Thus, algorithm  $\mathcal{A}$  yields an optimal budget allocation for BT, which by Lemma 5.4 yields a budget radius of at most  $2 \cdot \text{LB} \cdot \log^2(n) \leq 2 \cdot \text{BR}(G) \cdot \log^2(n)$ .  $\square$

## 6. Conclusion and Future Work

The paper introduces a new model for optimization problems on graphs. The suggested *budget* model was used to define facility location problems such as center and median point. For the tree case, optimal algorithms are presented for both aforementioned problems. For the general metric center point problem, an  $O(\log^2(n))$  approximation algorithm is presented. The new model raises a set of open problems e.g.,: i) Complexity: there is still a considerable gap between the hardness result and the approximation factor in the general case of budget radius. ii) Facility location: Find approximation algorithms for the k-center, k-median on general graphs. iii) Graph optimization: minimizing the diameter of the budget graph.



## References

- [1] A. M. Campbell, T. J. Lowe, and L. Zhang. Upgrading arcs to minimize the maximum travel time in a network. *Netw.*, 47(2):72–80, 2006.
- [2] V. Chepoi, H. Noltemeier, and Y. Vaxès. Upgrading trees under diameter and budget constraints. *Networks*, 41(1):24–35, 2003.
- [3] Victor Chepoi and Yann Vaxès. Augmenting trees to meet biconnectivity and diameter constraints. *Algorithmica*, 33(2):243–262, 2002.
- [4] M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley-Interscience, 1995.
- [5] S. T. McCormick Li, C. L. and D. Simchi-Levi. On the minimum-cost-bounded diameter and the fixed-budget-minimum-diameter edge addition problems. *Operations Research Letters*, 11:303–308, 1992.
- [6] N. Megiddo. The weighted Euclidean 1-center problem. *Math. Oper. Res.*, 8(4):498–504, 1983.
- [7] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [8] S. Nickel and J. Puerto. *Location Theory: A Unified Approach*. Springer, 2005.
- [9] A. Tamir. Improved complexity bounds for center location problems on networks by using dynamic data structures. *SIAM J. Discret. Math.*, 1(3):377–396, 1988.
- [10] J. Z. Zhang, X. G. Yang, and M. C. Cai. A network improvement problem under different norms. *Comput. Optim. Appl.*, 27(3):305–319, 2004.